1 Appendix

1.1 Dataset

We evaluate the performance of our proposed method using two public datasets for HMER: CROHME and HME100K.

1) CROHME [1] is a widely used dataset in the HMER task, captured using a Digitizing Tablet. The training set of CROHME consists of 8,836 images of handwritten mathematical expressions. The CROHME is originally an online HMER dataset, we use the online trajectory points sequence to generate the offline images. The CROHME also provides three test sets: CROHME 2014, 2016, and 2019, containing 986, 1,147, and 1,199 instances, respectively. The CROHME dataset includes 101 categories of characters.

2) HME100K is a public HMER dataset proposed by [2]. It contains a training set of 74,502 images and a testing set of 24,607 images. HME100K captures images from realistic scenes, incorporating various factors such as twists, blur, and intricate backgrounds. The dataset encompasses 245 categories of characters. In comparison to the CROHME dataset, the HME100K dataset closely aligns with real-world application scenarios and provides a significantly larger volume of data, both for training and testing. Consequently, the HME100K dataset is better suited to accurately assess the performance of the model.

1.2 Implementation Details

We categorized the experimental setup and specific hyperparameter descriptions into several major categories: data process, encoder, decoder, optimizer, and training. Within each category, we provided detailed listings of the relevant experiments and hyperparameter settings.

Data preprocess: For the CROHME dataset, we preprocess the images into binary form. For the images in HME100K, we process them in RGB format, following [3].

Encoder: We employ DenseNet as the encoder, encompassing 22 DenseBlocks. The grouth rate of DenseNet is 24 and the reduction rate is 0.5, following [4].

Decoder: For the decoder, both the L2R and R2L decoders are composed of 2 layers of GRU cells, and the hidden state dimension is set to 256. The embedding dimension of the character and relationship are 128 and 64 respectively.

Optimizer: We used the Adadelta [5] as the optimizer and adopted the cosine annealing strategy to update the learning rate. The learning rate increased progressively from 0 to 2 in the first epoch and gradually decreased to 0 by the final epoch.

Training: In the loss function of the SLM, the values of λ_1 and λ_2 in Equation **??** are set to 1 and 0.1 respectively. Additionally, the weight of the L2R and R2L branches in the BAT loss is equalized. The overall loss function can be expressed as:

$$Loss = L'_{\mathrm{SLM}} + L_{\mathrm{SLM}} \tag{1}$$

where L'_{SLM} and L_{SLM} represent the loss of the SLM from the R2L and L2R branches in the BAT. We also use dropout technique in the training, with the ratio of 0.1. During the training process, we set the batch size to 32, and all experiments were performed on a single NVIDIA 3090 24G GPU. To accommodate differences in data volume, we trained the model for 45 epochs on the HME100K dataset and 240 epochs on the CROHME dataset.

References

[1] M. Mahdavi, R. Zanibbi, H. Mouchere, C. Viard-Gaudin, U. Garain, Icdar 2019 crohme + tfd: Competition on recognition of handwritten mathematical expressions and typeset formula detection, in: 2019 International Conference on Document Analysis and Recognition (ICDAR), 2019, pp. 1533–1538. doi:10.1109/ICDAR.2019.00247.

[2] Y. Yuan, X. Liu, W. Dikubab, H. Liu, Z. Ji, Z. Wu, X. Bai, Syntax-aware network for handwritten mathematical expression recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 4553–4562.

[3] B. Li, Y. Yuan, D. Liang, X. Liu, Z. Ji, J. Bai, W. Liu, X. Bai, When counting meets hmer: counting-aware network for handwritten mathematical expression recognition, in: European Conference on Computer Vision, Springer, 2022, pp. 197–214.

[4] C. Wu, J. Du, Y. Li, J. Zhang, C. Yang, B. Ren, Y. Hu, Tdv2: A novel tree-structured decoder for offline mathematical expression recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, 2022, pp. 2694–2702.

[5] M. Zeiler, Adadelta: An adaptive learning rate method, Cornell University - arXiv,Cornell University - arXiv (Dec 2012).